1. Details about hyper-parameter tuning: All reviewers asked about the details of our "grid search" algorithm for hyper-parameters t and λ, which is omitted in our paper. So we describe it in Algorithm 1 and will add it in the revision of our paper. The grid search aims to find hyper-parameters that lead to a FLOPs reduction $F \in (F^* - \delta, F^* + \delta)$, where F^* is the target reduction and δ, which is 5% in our paper, is a permissible range. Our algorithm first performs a coarse search on λ, then carries out a fine search on t. Empirically, our algorithm needs to try out about 5 groups of hyper-parameters to stop. However, for each group of hyper-parameters we only train 30 epochs, rather than the 120 epochs in full training. So the computational overhead of our grid search is not too much in practice.

Algorithm 1 Grid Search of hyper-parameters for Polarization Pruning

```
Input: Target FLOPs reduction $F^*$ with a permissible range $\delta$.
Output: Hyper-parameter values $(t, \lambda)$, such that, after pruning, the FLOPs reduction $F \in (F^* - \delta, F^* + \delta)$.
Initialize $(t, \lambda) = (1.2, 1.0 \times 10^{-4})$
Train network with $(t, \lambda)$ for 30 epochs, and get the corresponding FLOPs reduction $F$;
// The coarse search stage fixes $t$ and searches for $\lambda$ such that $|F - F^*| < 2\delta$.
while $|F - F^*| \geq 2\delta$ do
    if $F - F^* \geq 2\delta$ then $\lambda \leftarrow \lambda/2$ else $\lambda \leftarrow 2\lambda$ end ;
    Train network with $(t, \lambda)$ for 30 epochs, and get the corresponding FLOPs reduction $F$;
// The fine search stage fixes $\lambda$ and searches for $t$ such that $|F - F^*| < \delta$.
while $|F - F^*| \geq \delta$ do
    if $F - F^* \geq \delta$ then $t \leftarrow t - 0.1$ else $t \leftarrow t + 0.1$ end ;
    Train network with $(t, \lambda)$ for 30 epochs, and get the corresponding FLOPs reduction $F$;
```

2. Comparisons to Uniform Channel Scaling (UCS), DeepHoyer and other pruning methods: We have added experiments to compare with UCS method, using the experimental setups as suggested by Reviewer#2. Moreover, we added the experiment on 70% FLOPs reduction, as Reviewer#2 suggested to experiment on a wider spectrum of FLOPs. The results are shown in the following table. Note that "NS" is the pruning method using L1 regularizer. The results, together with the results in our paper, show that *our method is consistently better than UCS and NS under a wide spectrum of FLOPs reductions*.

We also thank Reviewer#2 for pointing out a recent related work that we have not noticed before: DeepHoyer, which also focuses on regularizers for pruning. DeepHoyer uses pre-trained models before pruning with the regularizer, which leads to better results due to the extra pre-training epochs before pruning. For fair comparisons, we run DeepHoyer using the same experimental setup in our paper, which means exactly the same number of training epochs are used for all methods. The results show that, *compared to DeepHoyer, our method has much less accuracy drop under the same FLOPs.* Due to the time limit, we only add experiments on the CIFAR10/ResNet56 and ImageNet/MobileNet setups. It is reasonable to expect similar results on other dataset/structures and we will add them in the revision of our paper.

Model / Arch	Approach	Baseline (%)	Pruned Acc. (%)	Acc. Drop (%)	FLOPs Reduction
ImageNet / MobileNet v2	UCS	72.0	71.4	0.6	27%
	DeepHoyer (Our Impl.)	72.0	71.5	0.5	30%
	Ours	72.0	71.8	**0.2**	28%
CIFAR10 / ResNet56	UCS	93.80	93.43	0.37	50%
	DeepHoyer (Our Impl.)	93.80	93.54	0.26	48%
	Ours	93.80	93.91	**-0.11**	50%
CIFAR10 / ResNet56	NS	93.80	91.20	2.60	68%
	DeepHoyer (Our Impl.)	93.80	91.26	2.54	71%
	UCS	93.80	92.25	1.55	70%
	Ours	93.80	92.63	**1.17**	71%

As for other references listed by Reveiwer#2, our results are better than those in references [3,4,5]. And although some results in [2] and [6] are better than ours, the experimental setups of [2] and [6] are quite different from ours. Additional tricks such as label smoothing or longer training epochs are employed by [2] and [6] in their fine-tuning stage. And we did not use these tricks. In summary, **experiments show that our method does achieve the STOA in channel pruning**, and we are eager to perform fair experimental comparisons with any other channel pruning method to prove this.

3. Statistical stability of our experiment results: We only listed experiment results of single runs in our paper, as most previous works did. But we have performed multiple runs of experiments and find that our results are stable. We will add the confidence intervals of multiple runs of our results in the revision of our paper.

4. About pruning layers: Reviewer#4 pointed out that we only perform pruning on specific layers of networks. We think there is a little misunderstanding. We perform pruning on all layers. Although we didn't impose polarization on the "bn3" layers of ResNet, we still prune the "bn3" layers by eliminating neurons with scale factors close to 0.